JOHNSON GRANT
IN-61
7303
P-31

SPACECRAFT SOFTWARE TRAINING NEEDS ASSESSMENT RESEARCH
Final Report

(NASA-CR-188103) SPACECRAFT SOFTWARE TRAINING NEEDS ASSESSMENT RESEARCH Final Report (Houston Univ.) 31 p CSCL 09B

N91-20786

Unclas
G3/61 0007303

Shirley Ratcliff
Katharine Golas
Southwest Research Institute

April 27, 1990

Cooperative Agreement NCC 9-16
Research Activity No. ET.16

NASA Johnson Space Center
Engineering Directorate
Flight Systems Division

Research Institute for Computing and Information Systems
University of Houston - Clear Lake

T·E·C·H·N·I·C·A·L R·E·P·O·R·T

The RICIS Concept

The University of Houston-Clear Lake established the Research Institute for Computing and Information systems in 1986 to encourage NASA Johnson Space Center and local industry to actively support research in the computing and information sciences. As part of this endeavor, UH-Clear Lake proposed a partnership with JSC to jointly define and manage an integrated program of research in advanced data processing technology needed for JSC's main missions, including administrative, engineering and science responsibilities. JSC agreed and entered into a three-year cooperative agreement with UH-Clear Lake beginning in May, 1986, to jointly plan and execute such research through RICIS. Additionally, under Cooperative Agreement NCC 9-16, computing and educational facilities are shared by the two institutions to conduct the research.

The mission of RICIS is to conduct, coordinate and disseminate research on computing and information systems among researchers, sponsors and users from UH-Clear Lake, NASA/JSC, and other research organizations. Within UH-Clear Lake, the mission is being implemented through interdisciplinary involvement of faculty and students from each of the four schools: Business, Education, Human Sciences and Humanities, and Natural and Applied Sciences.

Other research organizations are involved via the "gateway" concept. UH-Clear Lake establishes relationships with other universities and research organizations, having common research interests, to provide additional sources of expertise to conduct needed research.

A major role of RICIS is to find the best match of sponsors, researchers and research objectives to advance knowledge in the computing and information sciences. Working jointly with NASA/JSC, RICIS advises on research needs, recommends principals for conducting the research, provides technical and administrative support to coordinate the research, and integrates technical results into the cooperative goals of UH-Clear Lake and NASA/JSC.

Spacecraft Software Training Needs Assessment Research

Final Report

Preface

This research was conducted under the auspices of the Research Institute for Computing and Information Systems by Shirley Ratcliff and Katherine Golas of Southwest Research Institute. Dr. Glenn Freedman, Director of SEPEC, served as RICIS research representative.

Funding has been provided by Flight Data Systems Division, Engineering Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA technical monitor for this activity was Robert N. Hinson, of the Software Development Section, Flight Data Systems Division, Engineering Directorate, NASA/JSC.

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of NASA or the United States Government.

Southwest Research Institute
6220 Culebra Road
San Antonio, Texas 78228

Project No. 05-3065

SPACECRAFT SOFTWARE TRAINING NEEDS ASSESSMENT RESEARCH

Prepared for:

NASA/Johnson Space Center

Prepared by:

Shirley Ratcliff
Katharine Golas, Ph.D.

April 27, 1990

EXECUTIVE SUMMARY

Requirement. The Flight Software Branch of the Spacecraft Software Division at the Johnson Space Center contracted with Southwest Research Institute to identify the problems, along with their causes and potential solutions, that the Branch configuration management analysts were encountering in performing their jobs. The SSD management intends to give these analysts the best training and information management support available in order to promote the highest quality performance in managing changes for the Spacecraft Software System.

Conclusions. The significant conclusions that emerged from the study were that sophisticated training applications would provide the most effective solution to a substantial portion of the analysts' problems. The remainder could be alleviated through the introduction of tools that could help make retrieval of needed information from the vast and complex information resources feasible.

Most management concerns derive from, or are related to, the loss of experience, skill, motivation, and an inability to grasp the relationship of individual functions to the big picture—institutional memory—which results from the retirement of veterans who have grown up with the project from its inception. Specifically, management sees the need for: (1) a way to shorten the learning curve for new analysts; (2) a means by which analysts can acquire a deeper working understanding of interrelationships within the big picture without experience in performing each function; (3) making explicit for new employees the various implicit rules and strategies, standards, and guidelines which have developed over time; (4) a process by which to evaluate individual employee's performance in an atmosphere which, by its nature, depends on self-supervision and which contains ultimate performance safeguards which can obscure individual contributions.

The analysts, for their part, express the same basic issues from a somewhat different perspective. They see a need for: (1) more job aids that structure needed information for ease of retrieval and application, including guidance on the availability of such tools that have been developed by other analysts; (2) a top-down functional breakdown of the big picture that would enable them to understand the interrelationship of the processes and provide a framework for the massive amount of detail they work with; (3) a means of comprehending the requirements of the various parts of the system without hands-on experience in performing all of the functions; (4) more time to learn and to plan their work.

Recommendations. Southwest Research Institute recommends a range of approaches to solving specific problems. Providing a job context for teaching processes and procedures will increase both interest and understanding, and can be done in a variety of modes. Recommended technology-based solutions include process and situational simulation, interactive video, hypermedia, data base management support, and performance support tools.

TABLE OF CONTENTS

TABLE OF CONTENTS

page

LIST OF FIGURES

LIST OF APPENDICES

1.0 INTRODUCTION

Analysts in the Flight Software Branch (FSB) of the Spacecraft Software Division (SSD) at the Johnson Space Center manage the software configuration for the maintenance of spacecraft software. Management within the FSB wants to adopt the best means of providing analysts with information or tools they need quickly and effectively, and they believe that supporting learning through the use of advanced hardware, software, and courseware can be the answer. During the course of this study, Southwest Research Institute (SwRI) researchers conducted interviews with each member of the FSB and found the analysts and managers to be very cooperative and helpful in responding to our questions.

Managers appreciate the motivation and competence of the analysts, and seek to utilize their capabilities to the fullest to enhance productivity and job satisfaction. It is also management's desire to fairly distribute the workload in a supportive team environment.

Analysts have firm opinions on the tasks they perform and the problems they encounter during job performance. Although always cooperative, analysts were very open about their concerns that tangible results from a needs assessment study may never get implemented.

To investigate the matter further, the Flight Software Branch contracted with the Training Systems and Simulators Department at SwRI to conduct this needs assessment study.

2.0 METHODOLOGY

2.1 Overall Approach

The training needs assessment approach called for a systematic analysis of tasks and performance problems identified in the Spacecraft Software Division, Flight Software Branch. The researchers collected data and opinions from the analysts, section heads, and branch chief using structured interviews and open-ended survey questionnaires. They also reviewed technical documentation and existing training courses, and observed the analysts in their work environment.

The following discussion provides an overall description of SwRI's approach to conducting the training needs assessment and task analysis. See Appendix F for a full description of the methodology.

2.1.1 Needs Assessment Strategy

SwRI conducted the Training Needs Assessment (TNA) research using a model recently published after eight years of research by Dr. Allison Rossette at San Diego State University. The work included the development of data collection instruments, and the analysis and summary of research findings. Rossette's model, referred to as a "purpose-based training needs assessment," is so entitled to distinguish it from other approaches, which do not provide a method for getting face to face with real performance problems. Rossette's purpose-based TNA model focuses on getting the information needed to make informed decisions and recommendations.

2.2 Interviews with Analysts

The study researchers conducted two sets of interviews, approximately two months apart, with the 14 analysts in the Branch.

In the first interviews, the research personnel asked for basic information on a number of issues.

- Background data about the interviewees
  - job title
  - number of years in present job
  - education
  - previous job experience
  - personnel with whom they frequently interact

- What job tasks they perform and what they do in these tasks

- What training is available to them

- What they think would help them do their jobs better

In most cases, responses to the structured questions led to an open-ended exploration of problems the analysts were having in performing their jobs and suggestions they thought might alleviate the problems.

During the second interview, the researchers administered task rating worksheets (see Appendix C for sample). The worksheet asked for responses from each of the 14 analysts on:

- frequency of task performance
- importance of the task
- task learning difficulty
- individual or team task performance
- current training for task performance

2.3 Interviews with Management

The interviews with FSB section heads and the branch chief took place approximately three months after the start of the study.

The interviews with management elicited information about the analysts' general skills and abilities, and general attitudes toward their jobs. Managers were asked:

- What they thought the analysts did best, and in what areas they believed they needed the most improvement

- What major constraints seemed to limit the analysts' performance

- What positions they had the most trouble keeping adequately staffed

- What job functions they believed were the most crucial

- Where they believed training was needed the most

- What they perceived were the major problems with the way the analysts currently managed the software maintenance process

3.0 RESULTS OF DATA COLLECTION

3.1 Overview of Current Situation

3.1.1 FSB Management Efforts to Meet Needs

FSB management's concerns with training needs, as demonstrated by the funding of this and other studies, show a commitment to quality of work. In an early interview, one manager stressed the importance of quality while maximizing accuracy and speed. Management's intentions are to pursue state-of-the-art training options as well as to streamline the work while also providing analysts with ways to rapidly get up to speed in the complex software management work environment.

Management has a supportive attitude toward providing the analysts with the help they need. Several analysts expressed appreciation for the motivation that management provides through a historical perspective of the flight software branch.

Since the basic nature of orbiter software has evolved from requirements development to maintenance, management tries to give new hires a realistic idea of what to expect on the job. The new hires understand that configuration management has them distanced from the code, so they do not become disillusioned about designing and implementing code once they start the job. On the other hand, they also understand that the job has very high NASA visibility.

FSB management requires analysts to give presentations in their functional areas to both management and coworkers. These presentations give the analysts an opportunity to pull together and document information about their work. The presentations are useful in familiarizing other analysts with the functional scope of the orbiter software and also aids future cross-training of backup tasks. See Appendix B for complete list.

Ongoing plans to address computer configurations, networks, and information management issues are relevant to training needs. Data base support makes training data more accessible and easier to handle.

3.1.2 Description of Available Training

3.1.2.1 Previous Needs Assessment Study

A report from a previous training needs assessment study (Seals, 1988) was reviewed. A flowchart of recommended courses and subject-matter hierarchy was produced, from which CBT training courses are being developed. Management considers the hierarchy to accurately represent the scope of topics concerning the orbiter software. The hierarchy is currently being evaluated by an analyst in the FSB.

3.1.3 NASA Training

There are a series of high-level NASA MOD courses. Content areas include an overview of MOD and how MSD support ties in. The Phase I MOD course takes three weeks to complete and covers technical aspects of shuttle/operations/station. It also gives a facilities overview of each division. "Summary of MOD Phase I Orientation" (April 27, 1988) rates the orientation classes as to how relevant the information is to analysts in the Flight Software Branch.

"MOD/FCOD Training Materials Library Document Inventory" (May/June 1989) lists and categorizes training material in the MOD/FCOD library. Many of the documents contain topics of interest to the SSD analysts.

"Existing SMAP Courses" and "Proposed SMAP Courses" outline the objectives of Software Management and Assurance Training courses.

DPSD instructor-led and independent study courses are available from the Education Coordinator, CSC/R20, or Independent Learning Facility. Other DPSD courses are available by videocassette, videodisc, and CBT. A publication entitled "DPSD Courses" lists the available classes, class code, number of hours, and teaching method.

3.1.4 On-the-Job Training

Although on-the-job training is not structured within the Branch (meaning that it is not a formal process using standard guidelines and documentation), many of the analysts reported OJT as their primary means of learning.

3.1.5 Contractor Sources

IBM provides training and development testing courses. Classes are currently offered using the IBM manual "An Introduction to the Orbiter Data Processing System" as a primary source of information. A single lesson from this course sometimes only covers one page from this manual. Many topics from the subject matter hierarchy are also interspersed throughout the lesson. This indicates that the knowledge necessary for the analysts is complex and tightly coupled. Valuable information can be derived from comparing and combining different sources of similar information, and determining how the information might be presented more clearly. Documenting discussions analysts have might be good indicators of subjects that analysts want more information on.

3.1.6 Computer-Based Training (CBT)

CBT courses have been prototyped by Barrios using the SAM IV authoring system, and the branch has developed two HyperCard™ CBT lessons referred to in this report as the FCOS Tutorial and the Sequencing Lesson.

3.1.7 SAM IV Courses

Documents (Barrios, 1989) that reflect CBT information content for Bus Reconfiguration and Guidance, Navigation, and Control were reviewed. Actual on-line CBT lessons which were developed by Barrios (GPC Synchronization and PASS-I-Load Reconfiguration) were not reviewed.

To develop the CBT lessons, the knowledge transfer between the technical contractors and the training authors is coordinated and enhanced by SSD analysts. Allowing the analysts to express their technical concepts on a flexible vehicle like HyperCard™ is more likely to directly generate creative representations of the information content than if the knowledge transfer is limited to textual transfer between SSD analysts, technical contractors, and training authors.

3.1.8 Evaluation of HyperCard™ Lessons

Two CBT lessons developed by NASA staff in HyperCard™, Process Management (Phillips, 1989) and Sequencing (Perrera, 1990), were reviewed to assess their instructional effectiveness and the effective use of HyperCard™ as a CBT authoring package/system. The subject matter experts are software engineers and do not have experience or expertise in instructional systems development, so the instructional strategies they used in the experimental lessons are not as good as they can be. Both lessons make good use of basic interactive programming techniques in terms of "user friendliness." For example, a student can quit at any point and have his or her position in the lesson saved. In the sequencing lesson, a return button on the tree card allows students to go back to the card they were previously viewing.

The acronym definition was cleverly implemented in both lessons; and in the sequencing lesson, an "unclear concepts" feature was also implemented. The "unclear concepts" feature allows a student to get a definition of any technical term simply by moving the mouse near the term. A definition of the term appears in a window.

The FCOS tutorial lesson uses graphics to illustrate ideas, which makes the lesson more interesting. Twice as many graphics are used in the FCOS tutorial as the Sequencing lesson. There is less text per page in the FCOS lesson than in the Sequencing lesson. Twenty-six percent of the cards in the FCOS lesson have graphics as well as text. Only twelve percent of the cards in the Sequencing Lesson have graphics as well as text. Features that the Sequencing Lesson improved or effectively utilized are:

- A user preference button allows users to skip preliminary quizzes, skip the HyperCard™ introduction, or check final quiz scores; this concept could be expanded to include other decisions that may be necessary.
- The tree card with a return button is an effective "map" that links the details to the overall concept.
- Before each preliminary quiz, a summary of each section clearly and simply states the information presented in the section. These summaries are so good, in fact, that we recommended they be revised slightly and presented at the beginning of each section as objectives. This will anchor the student and also provide trail markers that keep the student from getting lost in the information.
- Buttons and fields (return buttons and acronym expansions) with the same functions look and act alike. They should also appear in the same location on the screen. This consistency should continue between courses.

However, there is too much text in both lessons. A simple solution is to reduce sentence length and introduce more graphics. Information structuring aids, such as flow charts, matrixes, lists, etc., can help modularize information.

Recommendations for reducing the "learning curve" involved in generating the HyperCard™ CBT lessons include the use of computer-based tutorials on HyperCard™; however, one problem with generic tutorials is that they include information that is not necessary for the specific application (such as general programming techniques) while they may not include specific information which is needed (such as the use of "hidden fields" for the acronym and unclear concepts features). A customized tutorial that addresses how to implement certain features can reduce the HyperCard™ authoring learning curve. (See Solution for Performance Support Tools, Section 4.8.)

Also, use of existing shareware and public domain software aids can facilitate HyperCard™ applications by simultaneously decreasing the programming time and making the applications more sophisticated.

3.1.9 Work Environment

An SSD analyst manages the software configuration for the maintenance of spacecraft software throughout all phases of the software life cycle. The software areas include: guidance, navigation and control (GN&C); systems management (SM); vehicle utility (VU); primary and backup general purpose computer's (GPC) flight systems; payload; remote manipulator system; downlist; uplink; associated preprocessors for system software; and flight software compilers. In the process of managing software changes in their assigned functional area, an analyst reviews and researches requirements, participates in Board and internal meetings, controls publication of design documents, and prepares technical presentations. The test phase requires input, analysis, and approval of detailed verification test procedures and results. Rotating tasks have additional requirements, including the generation of other documents and other flight specific support. Analysts interact on a daily basis with primary (IBM) and backup (Rockwell) contractors, and paratechnical contractors (Barrios). They are also responsible for evaluating the contractor's performance.

Interviews with the analysts revealed the following categories of tasks:

- Process DR/CR
  - Review SW Requirements (CR/DR)
  - Research SW Requirements
  - Interface with Contractor Personnel (Disposition/Severity)
  - Attend Board Meetings (PASS & BFS DR)
  - Chair Boards
  - Attend Additional Board Meetings
  - Prepare FSW Mgmt forms
  - Interact with other Branch Members
  - Support Pre-SASCB
  - Support SASCB
  - Participate in working groups and mode teams
  - Produce Program Notes on Waiver's Document (each OI & Flight)

- Testing
  - Develop Scenario
  - Review Test Plan/Specification
  - Review Verification Test Procedures

SSD Interaction Process

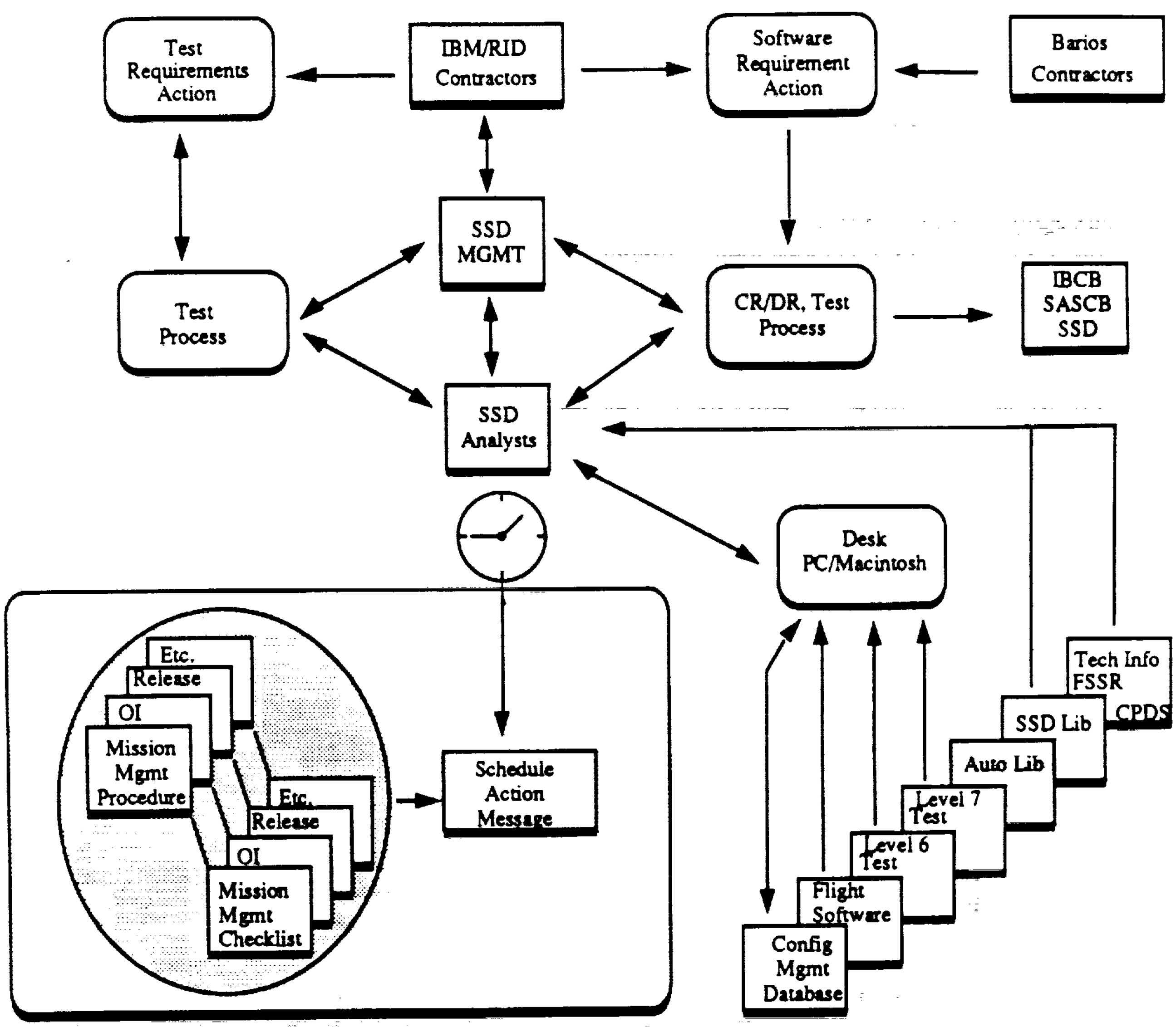

Figure 1 SSD Interaction Process

Review Verification Test Cases
Analyze Test Results

- Support Tasks
  - Principal Function Managers (IPL, DCP, BSS)
  - Certification Manager
  - DR Manager
  - OI Manager
  - Mission Manager
  - Backup/Substitute
  - Flight Support
  - Release Manager
  - T&O Manager
  - Produce Support Tools Document (once every OI)
  - Book Manager

- General Tasks
  - Control Documentation - Publish of Design Documents
    CPDS (IBM), PRD (Rockwell)
  - Maintain Current Professional Knowledge
  - Monitor Processes - Development Plan Review Meetings (Schedule Status & Integration Planning)
  - Attend CIP and CI Meetings
  - Evaluate Contractors
  - Develop Training/Presentations
  - Make Technical Presentations to Upper/Mid Management
  - Manage Facility

The task rating worksheet data revealed that the task performed most frequently by the analysts was "interact with other branch members." (The analysts reported that this task was performed once a day or more than once a day.) The research data also revealed that analysts spend more time in verbal communications than in nonverbal research techniques, such as reviewing printed technical documents or accessing information on-line. (Analysts reported that they reviewed SW requirements and researched SW requirements once a week to once a month.)

The analysts reported spending 30 percent of their time interfacing with contractors, which the analysts also rated as very important. They interface with technical contractors (IBM & Rockwell) concerning the software. Barrios provides support for software requirements processing and provides data base management. In addition to researching requirements, analysts review and approve test cases and results, where they interface only with technical contractors. The analysts interact via the telephone, electronic mail, paper documents, during meetings, and by direct discussions with individuals. Analysts require a lot of scheduling and time management skills to attend and assimilate the many meetings with coworkers, management, and contractors.

The analysts reported that analyzing test results is the hardest task to learn. The job which everyone agreed was performed on an individual basis was contractor evaluations.

Analysts can access a variety of written and electronic information. There is an abundance of detailed technical information the analysts review when they conduct their research on software requirements. Written data accumulates in manual form in an analyst's office, especially as it relates to their functional area. Additional manuals and related technical material exist in SSD's local library.

Configuration management data bases are available to the analyst on-line. The analysts receive information on the status of DRs and CRs. They upload information to these data bases as they process their requirements. They also check for concurrency (synchronize versions of a change) when they are presenting a CR or DR in front of a Board or SSD manager. Other on-line information access includes test information (levels 6 and 7) and the flight software code.

Figure 1 is an interaction flow diagram showing the general methods of interaction (analyst-individual, analyst-paper, analyst-meeting, analyst-computer). The rectangles represent sources of interaction; the rounded rectangles indicate a process involving multiple interactions. Shaded areas indicate something that is planned but not formally implemented.

3.2 Problems/Suggested Solutions

3.2.1 SSD Management Concerns

Because of the variety of skills required to manage software configuration, it is essentially impossible to hire "ready-made" analysts. As with many complex jobs, a manager considering hiring a new analyst balances trade-offs in the decision. Due to the low attrition rate in the Branch, the consequences of these decisions are critical. Some of the questions management deals with are:

- What can management do to facilitate learning for analysts that have difficulty acquiring new skills?
- What can management do to shorten the learning curve?
- How can management recognize symptoms before they become problems?
- How can management identify what deficiencies can be addressed by training?
- Can motivational problems be improved by training?
- Do the people need to be fixed or is the system the problem?

One problem in a job that requires a lot of self-supervision is that the source of outstanding work as well as deficiencies may be hard to identify. If a product is superior, it could mean either the producer's performance was excellent, the overseer's input made the difference, or both. How can management measure an analyst's performance and ability in a world where the checks and balances that safeguard the shuttle can obscure which analyst made a real impact? Some abilities, such as a commanding presence on a Board, can have their reward in visibility. Management needs tools to identify those who go the extra distance on their own when the work is not likely to be recognized.

In response to the question "in what areas do analysts need the most improvement," one manager responded that "they need to be independently motivated to study the FSW, and to hit the books and dig deep to solve problems." However, that same manager pointed out that there is no spare time really to do this; and to some degree, the analysts don't know how far to dig.

Another frustration management reported is when an analyst lacks the skills that management believes are entry-level requirements to do the job. In particular, some analysts lack basic writing skills. Management wants to find a means of improving performance in this area.

Another problem which management described is documenting the rules and strategies used to solve problems. An example is the decision making that goes into determining the severity and disposition of Discrepancy Reports. A small set of rules is frequently used and understood; however, a larger variety of guidelines exists in various analysts' heads to make less frequent decisions on severity and dispositions.

3.2.2 Retirement of Veterans

Historically, early configuration managers were involved in the entire software life cycle. They participated in creating the software and stayed on to maintain and update it. Current configuration managers lack the hands-on development experience. Management needs to know:

- What will happen when the senior work force retires?
- How can they give analysts twenty-five years of experience in the short time available?
- How can analysts get a working knowledge of the "big picture" without receiving first-hand experience in all the functional areas in the SSD branch?

3.2.3 SSD Analysts' Concerns

3.2.3.1 Accessibility of Job Aids and Other Information Support

The analysts generally agreed that the job aids, methods, and information currently available to them are useful. Many could not conceive of doing their job effectively without them. However, the analysts emphasized that there is not enough of this type of support available. There appears to be no systematic distribution of this information to new employees or co-workers. Finding job aids was often attributed to luck. Some analysts felt they had to invent their own job aids or strategies for processing information, even though a predecessor or co-worker may have struggled with the same problems and developed similar solutions. The job aids structure needed information for easy access; for example, Data Set Charts, library reference listings from MOD, or methods for notebook organization.

Analysts said that lack of direction for finding information resources was a major problem. They reported not knowing what resources were available or how to access what they knew existed. Numerous technical manuals are readily available, but the analysts often find the detail in them to be overwhelming, especially when they need to understand the software problem at a higher level. The analysts had trouble finding relevant information on procedural tasks, problem-solving tasks, and general overall subject matter. One analyst recommended a "lead engineer" concept, where responsibility for giving direction and technical support lies with the most experienced analyst (at the appropriate level).

3.2.3.2 Use of Job Aids

Seventy percent of the analysts reported that job aids would be the single best solution management could provide. A few analysts on their own initiative developed job aids while other analysts listed the following examples of where job aids could be developed:

- a list of experts, in NASA or in the contractor community, in particular functional areas
- a list of resources for researching software requirements
- written guidelines for implementing procedural tasks
- aids to learn technical jargon
- time management templates, which provide an idea of when to do things

3.2.3.3 Need for Top Down Approach

One analyst said it would be beneficial if a new employee could focus only on the task of processing a Discrepancy Report from start to finish, at a high level before having any of the details explained. This does not happen. They get their working knowledge of the spacecraft software from the bottom-up rather than the top-down. They learn all of the details for how to process a Discrepancy Report (DR), and only after they gain experience will they see how the process works and how it fits into the larger scheme of things. With a top-down approach, the system level functions and components are learned first; and this overall picture becomes the cognitive organizer for learning subsequent, more detailed information. The top-down approach, which is supported by cognitive learning theory, clusters and sequences information so that it constitutes a meaningful whole; both in itself and in relation to what the student previously learned about the system. The assimilation theory (Ausubel, 1968) suggests that new meanings are acquired by the interaction of new knowledge with previously learned concepts or propositions. This interaction results in a modification of both the meaning of the new information and the meaning of the concept to which it is anchored (Ausubel, 1978).

3.2.4 Information Overload

With the SSD analysts, once a difficult task or process is complete, the analysts have trouble extracting the useful lessons learned, quantifying them, and storing them away for later use. Even though some tasks are performed by following a set of sequential steps, the requirement to perform several related tasks concurrently makes documenting the process difficult. Additionally, if the complete task is learned on the job over a long period of time, the delay between performing the steps can slow the learning process.

3.2.5 Lack of Practical Experience and a Concrete Connection to the Real World

Suggestions from the analysts for overcoming the problem of information overload (defined here as the large amount of information as well as the complexity of the subject matter) were to provide realistic, "hands on" training in the best appropriate context. For example, what happens in a board meeting can be picked up by going to one (or through simulation). Once analysts have experienced this process, they are freer to think about how to determine dispositions, severity, and software implications. One analyst suggested that a prerequisite for analyzing test results should be first-hand experience with assembly language to get a better understanding of software/hardware interfaces. His point was that simple things may appear abstract and distant if the analyst doesn't clearly understand the types of events that can happen during a test run.

3.2.6 Lack of Time

Since most analysts are highly motivated, they can usually eventually figure out how to solve problems. However, most wish they had more background knowledge, more time to learn, and especially more time to plan. One analyst said that responsibility for innovation lies at the individual level, but the individual is usually too busy putting out fires to be innovative. One analyst suggested regularly scheduled brainstorming sessions (with only analysts allowed) as a good way to share information and suggestions for streamlining processes.

3.2.6.1 Lack of Knowing What They Need To Know

Both new and experienced analysts are uncertain over how much they need to know about an issue to satisfy the task at hand. The new employee has difficulty knowing what information is needed, whether it needs to be understood in detail, understood as a summary, or understood as an awareness. For the more experienced, this problem sometimes surfaces with the analyst pursuing a subject in too great a detail.

3.2.6.2 Absence of Formal Mentor Relationships

Thirty percent of the analysts recommended one-on-one mentorships between a new hire and an expert. One analyst mentioned that this would be especially useful when running the flight software. It should be noted that one analyst said mentorships would never work because there is never enough time.

3.2.6.3 Other Solutions

Analysts recommended numerous other solutions, including automating some of the forms and having group "jam sessions," where analysts can openly discuss problems and solutions. A few analysts recommended providing hands-on exposure to the testing facilities and simulator, and running a FEID job; developing a course on data sets; and isolating the DR process and going through the process from beginning to end. Finally, analysts recommended development of guidelines for how to evaluate contractors and specific training in how to interact with contractors.

4.0 RECOMMENDED SOLUTIONS

4.1 Introduction

SSD management intends to develop prototype applications and to adopt an incremental approach to solving performance problems with quality training. SwRI's recommendations are based on systematic analysis of the Branch's problems, taking into account effective training solutions developed previously and embodying the training developer's insights on potential training enhancements and possibilities.

Analysis of the problems cited by both management and the analysts suggests that the issues can be sorted into two major categories: (1) those having to do with lack of method for retrieving what is needed from massive amounts of complex information, and (2) those resulting from lack of method for providing overall as well as detailed understanding of the software processes of all the functional areas.

In the first category, we would include management's need to make explicit the rules and strategies used to solve problems and analysts' complaints over lack of knowledge of available resources, such as job aids and the general lack of organization of overwhelming amounts of detail.

In the second category fall management concerns over shortening the skill learning curve, enhancing analysts' motivation to study, and giving configuration managers a workable substitute for hands-on development experience with the software. It also covers analysts' problems with lack of functional understanding or context for all the job task details, difficulty in sorting out steps of concurrently performed tasks, and difficulty in time management.

Recommended solutions to the first category of problems fall into the general area of effective information management that can be provided by hypermedia and data base management support.

Recommended solutions for the second category include technology-based instructional systems to provide functional and detailed understanding via synthetic experience, such as interactive simulations.

SwRI offers in its solutions greater utilization of advanced technology now and in the future. One of the major problems in today's world is not a lack of information but being able to retrieve needed information from the overwhelming amount that exists. For the SSD analysts, the feeling of scarcity in the midst of plenty comes from their heavy information load, complex knowledge requirements, time critical schedules, unfriendly machine interfaces, and tedious safety checks and balances.

For the able and motivated student, learning can take place when needed information is made available (Keller, 1983). However, even highly motivated and capable students will benefit from careful structuring of information, as illustrated in Figure 2. The speed and depth of learning can be improved by increasing the sophistication of the training and providing easier access to needed information.

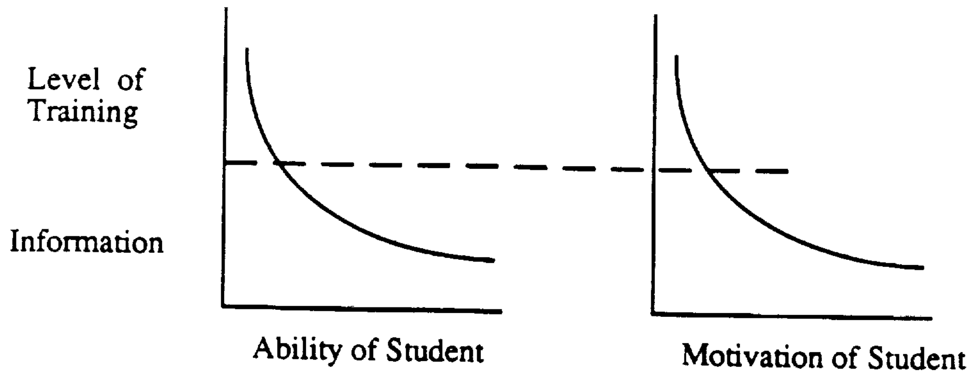

Figure 2 Structuring of Information

4.2 Teach in a Job Context

Problem: Distance between Software Knowledge Training and Job Context

Solution: Teach Intricacies of Flight Software in the Job Context

Training and presentations for SSD analysts, such as CBT lessons and IBM's Orbiter Data Processing System class, present knowledge of the software and hardware from a traditional classroom perspective, but without formal testing. For SSD analysts, the real test is generally to apply such knowledge to the job of software maintenance management. This requires more than a familiarity with the system; it requires a detailed understanding of how the software functions.

For maximum understanding and retention, this traditional approach works best when training classes occur near the same time as a relevant Discrepancy Report; immediate participation acts to anchor the knowledge. The approach also works well for the analyst who prepares a presentation because participation in the preparation of the presentation fosters retention and requires a deeper understanding of the material. Analysts who participate as students gain useful information, but they can receive and retain more information if more of the relevant skills and knowledge were taught in a job context. Putting flight software information in a job context also heightens motivation for both analysts and the audience.

Introducing an analyst, for example, to the bus management structure in the context of a relevant Discrepancy Report can heighten his motivation to learn. The presentation can continue to provide a detailed overview of the software structure, architecture, and functional modules as well as convey subtle information about how to do a DR. The presentation would then enable new analysts to model after the DR processing strategies they observe, and more experienced analysts to gain new insight, while learning more about the software structure. Finally, performance-related information generates more active participation by the presentation listeners.

Example: SSW Training Class - Mapping the Requirements to the Code

In a current SSW class (modeled after the IBM static analysis for revalidation), analysts investigate a functional area by mapping the requirements to the code from a list of functional modules and a list of documents that contain related information. This assignment has proven very effective in the past; it could be made more job relevant by:

- Having the analyst create a "DR" scenario that could provide a focus for the presentation.

- Giving more examples of charts that represent the functional flow of information or the coupling that occurs between modules. For example, a matrix can list critical phrases from a requirements specification vertically and software modules horizontally across the top. The student then places a check on the resulting grid for every functional module that plays a part in satisfying the requirement phrase. If the chart becomes too cumbersome, it can be presented at a higher level or broken into logical parts.

- Standardizing symbols and guidelines for data flow diagrams, as well as the traditional flow charts, and encouraging the analysts to present several alternative views of the same software functions.

4.3 Organize with Technology-based Solutions

Problem: Information Overload

Solution (1): Utilize Hypermedia

Hypermedia is a new technology that supports easy, unlimited association of information. Hypermedia also supports other modes of information presentation than mere text. Hypermedia is the best means for storing and delivering complex and a large quantity of information. Its associative characteristics also make it possible to be used as an on-line reference when the learning phase is over. These features make hypermedia an excellent medium for SSD's training needs.

Many aspects of an SSD analyst's job could benefit from on-line references and on-line job aids, ranging from cross-training and backup needs. Given the nature of the analysts' highly individualized schedules and the availability of networked computers, hypermedia is a viable solution.

SSD analysts are capable and motivated and should be able to learn from well-structured information. SSD has used its subject matter experts to develop internal training materials with Apple's hypermedia software, HyperCard™. The results show that this is a workable approach. It took the experts a minimal amount of time to learn the software and develop the training programs.

To obtain maximum benefits from having the analysts share their subject matter expertise, SSD should use instructional systems experts to develop standards and templates for computer-based information/training programs, and provide training in using these. For critical or complex content areas that call for more sophisticated instructional treatments, such as practice, feedback, test, and simulation, SSD should use instructional systems experts in collaboration with SSD's subject matter experts to develop the training programs.

Solution (2): Use a Relational Data Base to Organize and Store Attributes of Available Subject Matter

A structured method of organization is critical for enhancing modularity, reducing redundancy, recycling information, enhancing rather than reinventing information, and, most importantly, finding the most logical representation of the information that will facilitate learning. Supporting information should be organized modularly to facilitate changes. Stand-alone data base management systems (e.g., Oracle) can support the process of structuring such information that can be used as a resource for a variety of applications. Similarly, a library system like AUTOLIB, with careful attention key words and other stored references can also structure the information to support training development.

Example: DR Dispositions

DR dispositions have many different types of information associated with them. The definitions describe what the disposition codes mean. Often, there are written examples that illustrate the meaning of a disposition. There are strategies and rules that tell how an analyst arrives at a particular disposition. There are procedures that outline the order of events that must happen to process a DR. Graphics may exist that show a visual representation of the process. All of these types of information contribute to the analyst's understanding of DR dispositions. Once information is meaningfully categorized and stored in a relational data base structure, it becomes accessible and reusable.

Relational data base support gives the analysts the power to view a history of DR dispositions and add to that history for future reference (and training). Also, well-organized information on the DR process would benefit educational presentations that analysts are required to give. The DR process affects many groups at NASA, but most are only aware of their own perspective. Documenting other groups' perspectives of the process will help the analysts in future presentation preparations and enhance the analysts' general knowledge of the DR process.

4.4 Develop Process Simulations

Problem: Difficulty in Understanding Details Without a Top-Down Context

Solution: Utilize Interactive Simulation

In an educational context, a simulation is a powerful technique that teaches about some aspect of the world by imitating or replicating it (Alessi, 1985). Simulations encourage active learning by demanding student participation, and they are efficient both logistically and instructionally. Simulations enhance motivation by allowing the student to be an active participant in the learning situation. Usually what is learned in a simulation transfers well to the real world. As simplifications or models of real life, simulations make it possible to focus on or control key variables, and to repeat scenarios quickly and efficiently. Moreover, simplification can be instructionally advantageous. For example, research shows that a person will generally learn faster if details are eliminated at the beginning of the instruction. As the student becomes increasingly competent in dealing with the simple case, the simulation may then add detail to bring the student closer to reality.

Interactive Procedural Simulations

The purpose of most procedural simulations is to teach a sequence of actions that constitute a procedure. In such simulations, whenever a student acts, the computer program reacts, providing information of feedback about the effects the action would have in the real world. Based on this new information, the student takes successive actions and each time obtains more information. A primary characteristic of procedural simulations is that there is usually a correct or preferred sequence of steps that the student should learn to perform. However, there may be many different ways of reaching the same conclusions, not all of which are equally efficient. A well-designed procedural simulation provides the student with the opportunity to explore these different paths and their associated effects.

The newest technology for creating realistic computer-based simulations is Digital Video Interactive (DVI). DVI takes advantage of the digital storage capacity and data rate of the standard CD-ROM drive to provide features that are well beyond the capabilities of analog interactive videodisc (IVD). DVI has the ability of display up to one hour of full motion video from compressed digital data stored on a single, standard CD-ROM disk or on the hard disk on the computer. DVI's full motion video can be combined with foreground video objects, text, 3-D graphics and animation, and multitrack audio, all under the user's control.

Example: DVI Ascent

It appears that some of the content of the current HyperCard™ sequencing lesson would lend itself extremely well to interactive video delivery (Digital Video Interactive or Authorware). With interactive video, visually dynamic video footage of prelaunch and ascent guidance phases could be enhanced with graphics and animation sequences to make abstract information visually concrete.

A student interacting with a DVI simulation can solve problems, learn procedures, come to understand the characteristics of a certain phenomenon, and learn what actions to take in different situations. In each case the purpose is to enable the student to build a useful mental model and to test it safely and efficiently. In a DVI simulation the student learns by performing certain actions in a context that is similar to the real world. DVI simulations can be used to present initial information, guide the student in acquiring the information or skills, provide practice to enhance retention and fluency, and finally assess what the has student learned.

Example: The Test FEID Process

Analysts must interpret output listings from Test runs as part of their job even though their understanding of the process may be limited. They would be better able to handle the process, as well as their interactions with test contractors, if they became more familiar, through simulations of the test process, with the types and sequence of events that happen at the Test facilities.

Example: Use DVI to Teach Software Review Methods

Code inspection is a type of reviewing methodology that has proven itself in software design and planning. Fagan's inspection method has regularly measured increases in productivity on the order of 25 to 35 percent. More impressive gains result from using Code Inspection methods to validate the quality of the testing phases (Gilb, 1988).

Software engineers that are specifically trained in this formal technique can utilize it for software management. Training for this formal process lends itself well to interactive simulation on a computer-based platform.

4.5 Develop Situational Simulations

Problem: Analysts' Difficulty in Interacting with Contractors

Solution: Affective Domain Training

Situational simulations deal with the attitudes and behaviors of people in different situations, rather than with skilled performance. Situational simulations can be used to train students to meet performance objectives in the affective domain, for example to train SSD analysts how to interact and communicate with contractors. Unlike procedural simulations, which teach a set of rules, situational simulations usually allow the student to explore or test the effects of different approaches to a situation, or to play different roles in it. By going through a situational simulation a number of times, the student can learn a set of behaviors which will help to improve the probability of correctly interacting in an actual situation.

Example: Contractor Interaction in the Test Environment

The role as a software configuration manager has awkward moments, especially for a new analyst. Software testing contractors frequently have more technical experience and knowledge in the test environment than the SSD analysts. Video simulation can model positive professional behavior that deals with intimidating or awkward situations.

4.6 Provide Written Schemas

Problem: Lack of Written Rules and Strategies to Solve Problems

Solution: Provide Explicit Models of Rules and Strategies

Workers naturally default to a discovery learning method when the skills and knowledge that make up performance strategies are not explicitly taught. Although long-term retention of information is enhanced through discovery learning, it takes much longer to master the needed skills.

SSD management encourages analysts to formally document procedures associated with CR and DR management, and rotating roles such as OI, Mission, and Release management. This lends structure and direction to those job roles and helps target possible areas for automation. However, linear procedures do not capture the expertise that enables an analyst to understand the decisions and inputs from one step to the next. These decision strategies are difficult for many experts to explicitly reconstruct and verbalize, especially when the decision branching is complex. Other experts are more adept at reconstructing and communicating their strategies. These experts are the best source for creating an explicitly stated model for skills and strategies related to performance of particular tasks. Instructional designers are trained to work with such experts to construct and refine the models. Having this collective experience in writing will greatly enhance the transfer of expertise.

Example: OI Manager

One experienced analyst mentioned the frustration associated with helping another analyst document detailed procedures for OI management. There are obvious steps that warrant instructions, such as how to integrate charts and output from software applications that support the weekly status report. Other OI procedures are already documented at a high level. The analyst mentioned that an OI manager could document a sequence of problems for one OI but didn't think a very large percentage of the problems would map very well into the next OI manager's problems. Further questioning revealed that the analyst's own performance of rotating tasks depends on developing organizational strategies tailored to the job role. This involves designating notebook categories and methods of tracking related information. As the "system" evolves, the analyst is able to manage the complex variables related to performance.

Systematic development such as this can form the basis for teaching strategic models. Research has shown that explicitly describing the reasoning behind the method, and then associating the skills and knowledge needed to perform the procedures can shorten the learning curve for job performance. Valuable time is lost if each analyst must redundantly discover and develop their own strategies. Once a methodology is defined, scenarios that are unique to a particular OI then become a tool for implementing the model.

4.7 Implement Job Aids

Problem: Lack of Knowledge of Resources

Solution: Create and Distribute More Job Aids

Job aids contribute to job performance by supplying the following types of information: guidelines, open-ended prompts or a structured format, and organized data. Many provide passive information, whereas others are more actively integrated into an intermediate step or final task product. Procedure checklists, task checklists, worksheets, and other job aids provide cognitive organizers that allow the analysts' to be more productive in performing the task. Improvements made by one analyst can be shared by everyone, and seeing good job aids can trigger ideas for better ones.

Integrating collective ideas for performing a recurrent task starts with developing a form. Fourth generation language support such as Oracle Sql-Forms provides the capability to develop fast prototypes that are relatively easy to change. Producing and updating a prototype costs a fraction of what it would take to develop traditional software applications that create and maintain forms. (See Appendix D.)

Example: Contractor Evaluation Worksheet

Design a worksheet that supports contractor evaluations. In addition to the current categories of generic, general, and specific evaluation criteria, the form can restructure some of the guidelines in the format of rating worksheets, short answers, and additional comments. Scaled checklists can normalize some of the qualitative decisions with rating categories such as: below average, average, above average. This provides a dual benefit from an individual analyst's perspective and management's perspective as they collectively integrate several analysts' evaluations into one final contractor evaluation. Analysts who write initial evaluations can be consistent in their comparisons over time, while managers can better interpret a group of evaluations if the supporting data is standardized.

In addition to providing flexible prototypes that are relatively easy to maintain, a sophisticated tool for building a simple worksheet can lead to automating repetitive tasks. Worksheets that start out as pencil and paper tools can eventually serve as components of a near paperless workplace. Automation of repetitive tasks can eventually supply on-line access and input; individual ticklers; and filtering, cross-referencing, and tracking.

4.8 Other Considerations

Problem: Lack of Familiarity with Training Development Technology

Solution: Utilize Performance Support Tools and Other ISD Job Aids

There is a need to develop job aids to facilitate course development. A computer-aided device that permits automation of the analysis, design, development, implementation, and evaluation of training does not exist. However, a nonexpert instructional developer can use job aids and computer-based tools to facilitate the design and development process.

Puterbaugh, Rosenberg, and Sofman define a performance support tool as

> *Software designed to improve worker productivity by supplying immediate on-the-job access to integrated information, learning opportunities and expert consultation—with scope and sequence controlled by the user.*

Regardless of the simplicity or sophistication of development resources (time, money, designer expertise, development hardware and software), clear and documented guidelines can assist an incremental building of courses. If course design responsibility shifts between analysts and contractors, job aids can provide continuity. They can shorten the learning curve for new training authors and developers, and shorten overall development time for courses. The content and depth of the supporting guidelines and job aids are dependent on the training development resource constraints. Specific examples that job aids can support are:

- Establish Procedures for Course Development

  Examples:
  - Tutorials for current authoring system
  - Principles to follow for classifying and structuring information
  - Guidelines for interviewing and getting information from subject matter experts

- Provide Checklists

  Examples:
  - List of authoring topics and expected learning curves
  - Course evaluation checklists
  - Criteria for practice feedbacks

- Establish Communication Tracking Rules (related to training)

  Example:
  - Create a profs account for training and have analysts/managers forward related correspondence or send messages to this account

- Provide Format Control of Productivity Aids

  Example:
  - Save redundantly-used objects in a library (models, templates, graphics, objects)

- Provide Hints for Creatively Utilizing the Training Medium

  Example:
  - Use hypermedia, spatial representations, analogies